REMARKS

This application has been carefully reviewed in light of the Office Action dated May 20, 2004. Applicants have amended claim 8. Reconsideration and favorable action in this case are respectfully requested.

The Examiner has rejected claim 8 under 35 U.S.C. § 112. Applicants have amended the claim as suggested by the Examiner.

The Examiner has rejected claims 1-19 under 35 U.S.C. §102(b) as being unpatentable over U.S. Pat. No. 5,838,583 to Varadarajan. Applicants have reviewed this reference in detail and do not believe that it discloses or makes obvious the invention as claimed.

To anticipate a claim under §102(b), Varadarajan must show *every* element of the claim. Applicants strongly disagree that Varadarajan shows every element of the claims, nor does Applicants agree that the claims would be obvious based on Varadarajan.

With regard to claim 1, the Examiner continues to contend that Varadarajan teaches "assigning a fixed status to the datapath cells to prevent movement of the cells" in column 7, line 65 to column 8, line 3 and column 15, line 45 to column 16, line 7. Applicants do not agree.

The text starting at column 7, line 65 states as follows:

> The tile files 311 and the tile file list 313 specify sufficient information for the datapath placer 130 to determine the structural hierarchy in the datapath region 303. This allows the datapath placer 130 to place the datapath functions 309 in a manner that preserves the structural regularity in the datapath region 303, thereby yielding improved performance for the integrated circuit.

Nothing in this passage suggests that the datapath placer assigns a fixed status to the cells in the datapath region. It only states that structural regularity is preserved in the

datapath functions when placed by the datapath placer 130. There is nothing to suggest that the datapath functions could not be moved at a later time by the standard cell placer 170 within the bounds of the structural hierarchy.

The matter of applying a fixed status through use of cluster constraints has been discussed previously. The Varadarajan reference explicitly describes a cluster constraint as maintaining the relative position of two or more datapath functions in a datapath region with regard to *each other*; however, the clustered datapath functions are not fixed within the datapath region, nor are they fixed with regard to other datapath functions in a datapath region.

In the passage cited by the Examiner, Varadarajan states:

> However, in datapath region 1101, datapath functions 1109, 1110, and 1111 have been grouped into cluster 1113. Here, the datapath placer 130 must treat this cluster 1113 as a unit, *and can move this unit relative to datapath functions 1108 and 1112 only*. The ability to cluster datapath functions and thereby control the relative ordering of datapath functions within the region is useful for controlling timing between selected datapath functions. For example, there may be critical timing requirements between datapath functions 1109-1111 that the circuit designer needs to control. Without clusters, as in conventional systems, there is simply no way for the circuit designer to control the grouping of datapath functions in this manner, and thereby optimize performance as desired and enabled by the present invention. [emphasis added]

Thus, in the example state above, even though datapath functions 1109, 1110 and 1111 have been grouped into a cluster 1113, that cluster can move relative to other datapath functions (i.e., 1108 and 1112) in the datapath region. Consequently, clustering constraints do not prevent movement of cells, as stated in claim 1.

In the Remarks section, the Examiner states that Varadarajan also discloses fixing datapath function relative to standard cells in an embodiment shown in Figure 9 and column 14 lines 20-38.

The subject matter of this section of Varadarajan is not a separate embodiment. Figure 7 illustrates an outline of a IC 331 with instances not yet associated with any datapath function. Figure 8 illustrates the association of some of the instances of Figure 7 into datapath functions. Figures 9a-d illustrate the association of datapath functions from Figure 8 into a datapath region (dpath1). Figure 9d shows that whatever cells are not defined as part of a datapath region are considered to be non-datapath (standard) cells which will be placed by the standard cell placer. The cells in a datapath region, however, are not assigned a status of "fixed"; to the contrary, they can be moved by the datapath placer.

Further, as discussed in previous office actions, claim 1 requires that the routing of the datapath cells occurs *after* the step of transferring desired criteria regarding the other cells to the place and route tool and optimizing the layout based on said desired criteria, such that the datapaths cells are unmoved as different layout iterations are performed on the other cells.

The Examiner claims that this is taught in Varadarajan with the following passage found at column 3, lines 25-28, set forth below:

> Accordingly, it is desirable to provide a placement system that allows the circuit designer to cluster datapath functions, define constraints for net exits, and control the aspect ratio over the cell routing of the datapath functions.

This passage is concerned only with the layout of datapath functions, not the placement of standard cells. At no time does it suggest that the placement and optimization of standard cells (non-datapath function cells) occurs prior to routing.

On the other hand, Varadarajan *explicitly* teaches that routing of the datapath functions occurs prior to standard cell placement. The flowchart of Figure 2 of Varadarajan clearly shows that datapath placing *and routing* (block 208) is performed

prior to standard cell placement (block 205). The flowchart of Figure 2 is *unambiguous* as to this point.

It would appear that datapath functions are not "fixed" until the place and route datapath function 208 in Varadarajan is performed. At no time are they given a *fixed status*, as required by the claims. Further, if the Examiner is correct in the argument that the datapath functions are not routed until after the standard cells are optimized, then the datapath functions are not even fixed at the time of standard cell optimization.

Applicants therefore disagree with the Examiner's contention that Varadarajan anticipates the present invention. Accordingly, Applicants request allowance of claim 1 and dependent claims 2-7 and 16-17.

Additionally, for reasons stated above, Applicants request allowance of claim 8 and dependent claims 9-15 and 18-19.

In addition to the arguments set forth with regard to the independent claims, Applicants disagree that Varadarajan teaches the subject matter of dependent claims 16-19. In these claims, datapath cells are placed within matrices of slots ordered in rows and columns. Unused space may be provided between rows or columns (or both) such that open space in provided in the matrix (best shown in Figure 4 of the present specification). The space may thus be provided throughout the matrix for the remaining cells. By adjusting column and row spacing, free space 38 can be planned within the matrix to allow timing-driven placement of embedded standard cells along with the structured placement cells.

The Examiner cites columns 16 and 17 for showing this feature; Applicants assume this is the "interleaving functions" starting at line 55 of column 16. The function of interleaving in Varadarajan is to eliminate wasted space due to datapath functions requiring more over-cell tracks than would be otherwise available. Function interleaving

specifies for each datapath region, the number of rows per bit to be used by the datapath placer and routing space estimator.

In the example of Figure 14a-b of Varadarajan, Figure 14a illustrates an example where one row per bit is used – hence bit0 (B0) of functions F1-F5 use a single row and bit1 (B1) of functions F1-F5 use a single row. B0, however, requires 13 over-cell tracks, while only ten over-cell tracks are available over a single row. Accordingly, the datapath placer and router will be forced to add space between the rows to provide the additional three over-cell tracks to B0. This space will not be useful for anything else, since its overhead tracks will be used, and hence the space is "wasted." By allowing two rows per bit via the interleaving function, as shown in Figure 14b, twenty over-cell tracks will be available to B0, and the space between rows can be eliminated.

This space between rows in Figure 14a is unintended wasted space, not "free space between slots for datapath cells in which ones of said other cells are placed". Therefore, Varadarajan does not show the subject matter of these claims.

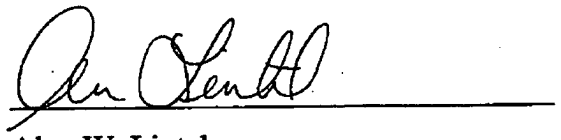
An extension of one month is requested and a Request for Extension of Time under § 1.136 with the appropriate fee is attached hereto.

The Commissioner is hereby authorized to charge any fees or credit any overpayment, including extension fees, to Deposit Account No. 20-0668 of Texas Instruments Incorporated.

Applicants have made a diligent effort to place the claims in condition for allowance. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Alan W. Lintel, Applicants' Attorney at (972) 664-9595 so that such issues may be resolved as expeditiously as possible.

For these reasons, and in view of the above amendments, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,



Alan W. Lintel
Attorney for Applicants
Reg. No. 32478

September 20, 2004
Anderson, Levine & Lintel
14785 Preston Rd.
Suite 650
Dallas, Texas 75254
Tel. (972) 664-9595